FIG.1

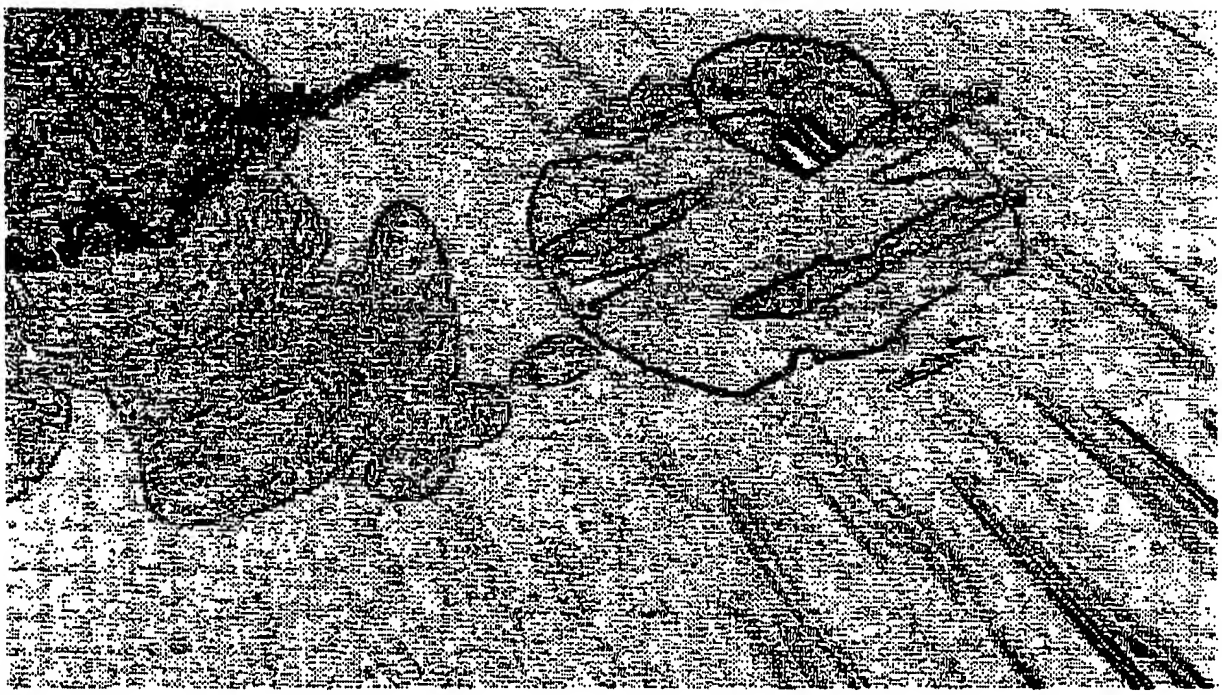


FIG.2

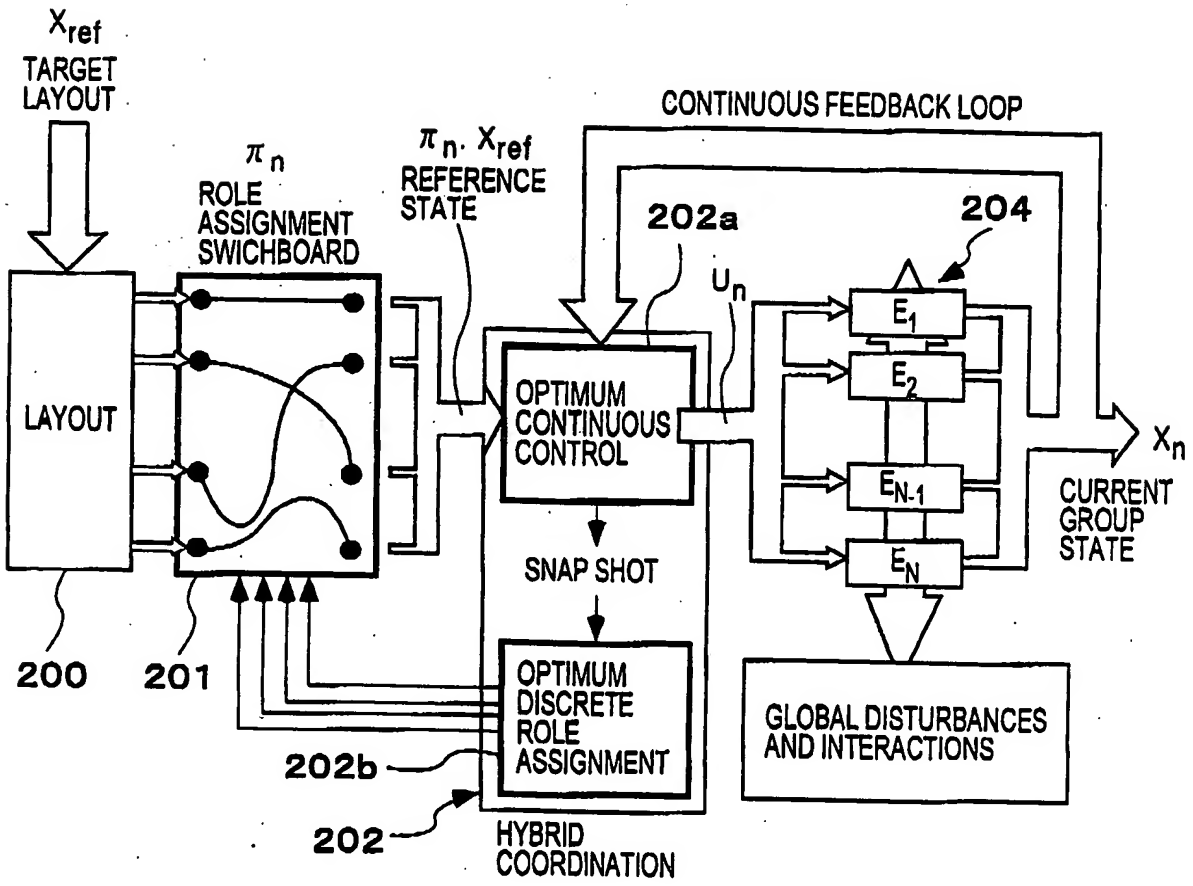

X_{ref}
TARGET LAYOUT
π_n
ROLE ASSIGNMENT SWICHBOARD
$\pi_n \cdot X_{ref}$
REFERENCE STATE
CONTINUOUS FEEDBACK LOOP
202a
204
U_n
E_1
E_2
E_{N-1}
E_N
LAYOUT
OPTIMUM CONTINUOUS CONTROL
X_n
CURRENT GROUP STATE
SNAP SHOT
200
201
OPTIMUM DISCRETE ROLE ASSIGNMENT
GLOBAL DISTURBANCES AND INTERACTIONS
202b
202
HYBRID COORDINATION

FIG.3

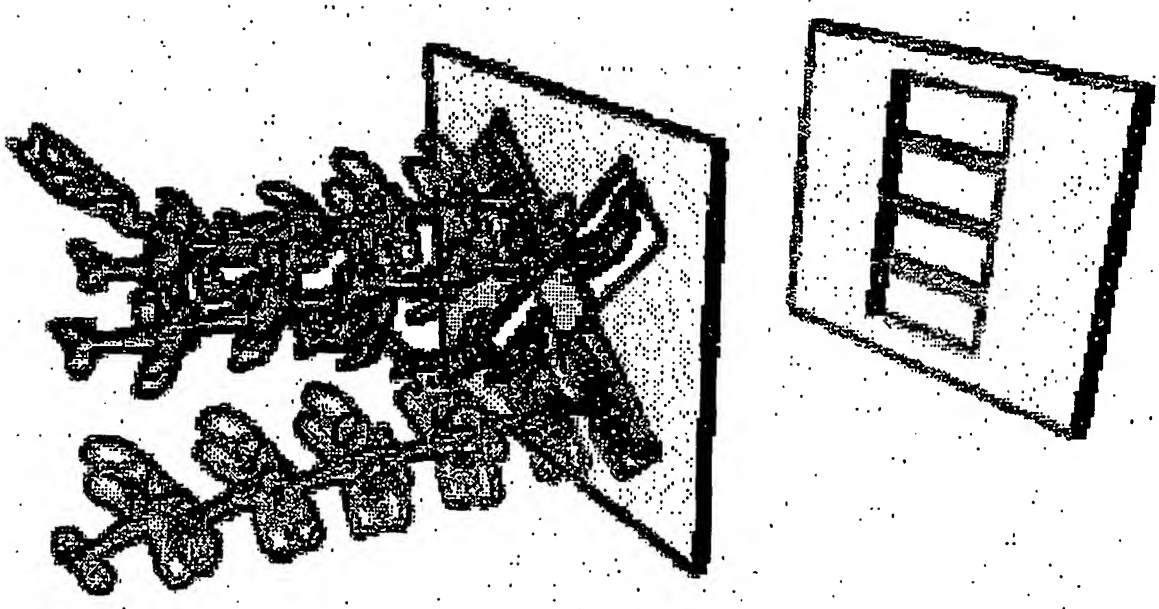


FIG.4

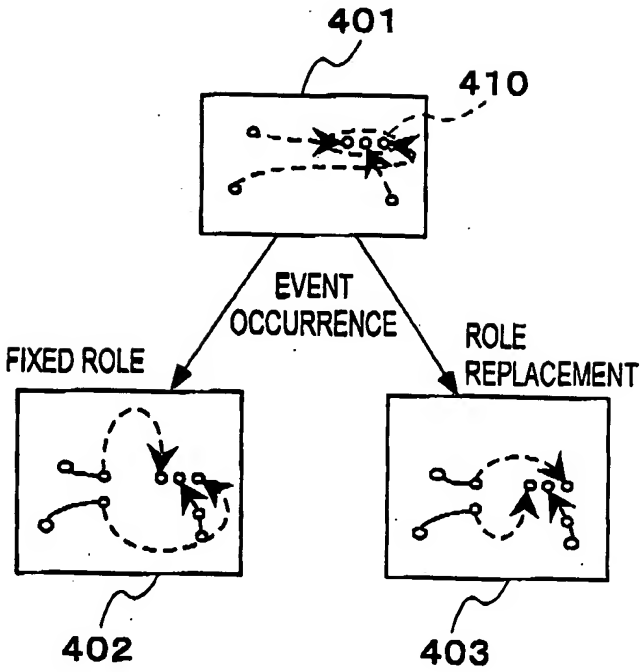

401
410
EVENT
OCCURRENCE
FIXED ROLE
ROLE
REPLACEMENT
402
403

FIG.5

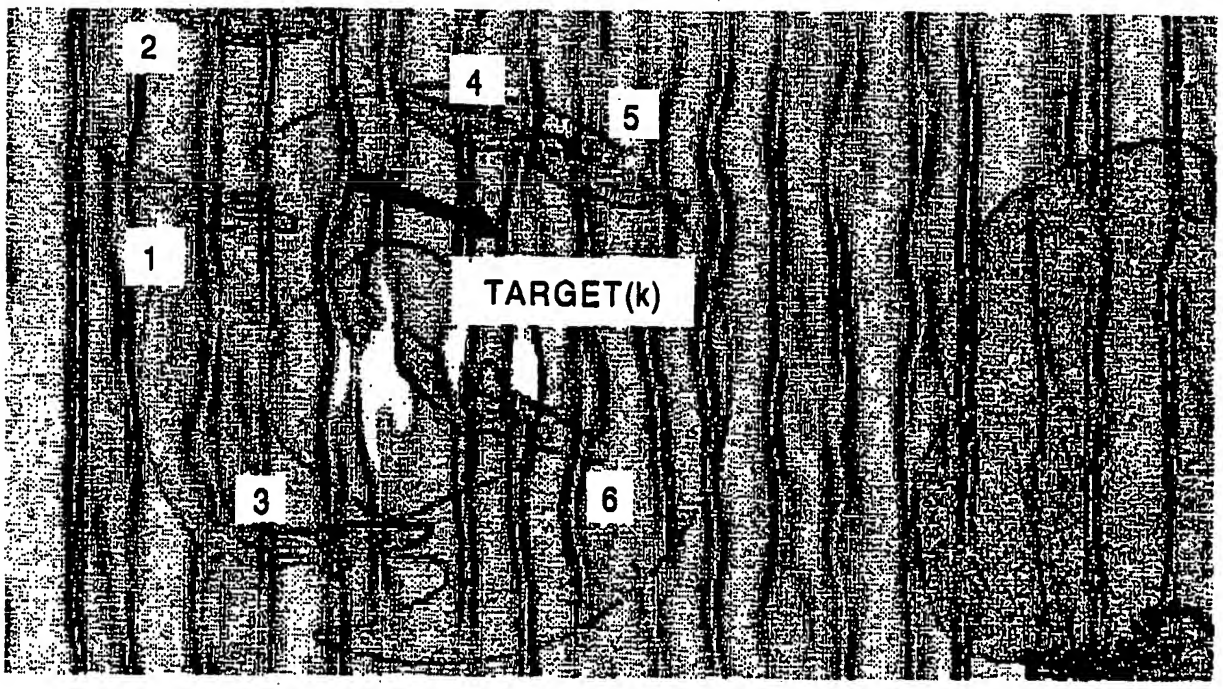


FIG.6

RUN-TIME COORDINATION ALGORITHM (A1)

EXERT CONTINUOUS MULTIVARIABLE CONTROLS (ROLE ASSIGNMENTS DEFINED BY π_n)

1. $U_n \leftarrow -G[X_n - \pi_n . X_{ref}]$

DYNAMIC SIMULATION STEP

2. $X_{n+1} \leftarrow f(X_n, U_n)$
3. $J_\infty \leftarrow (X_n - \pi_n . X_{ref})^T S_\infty (X_n - \pi_n . X_{ref})$

SORT ENTITIES ACCORDING TO THEIR CONTRIBUTION TO OVERALL COST

4 For each entity j do

5 $X_n^{-j} \leftarrow [X_n^1, X_n^2, \ldots, j^{th}$ coordinate of $(\pi_n X_{ref}), \ldots, X_n^N]$

6 $J_j(n) \leftarrow (X_n^{-j} - \pi_n . X_{ref})^T S_\infty (X_n^{-j} - \pi_n . X_{ref})$

7 Endfor

8 Sort entities by decreasing $J_j(n)$

LOOK FOR OPPORTUNISTIC ROLE SWAPS IN K WORST PERFORMERS

9 For $K!$ permutations $\pi(k)$ of K candidates do

10 $J_\infty(\pi(k)) \leftarrow (X_n - \pi(k) . X_{ref})^T S_\infty (X_n - \pi(k) . X_{ref})$

11 if $(J_\infty^* > |J_\infty(\pi(k)) + \Delta_{min}|)$ then

12 $J_\infty^* \leftarrow J_\infty(\pi(k))$

13 $k^* \leftarrow k$

14 endif

15 endfor

UPDATE ROLE ASSIGNMENT SWITCHBOARD

16 $\pi_{n+1} \leftarrow \pi(k)^*$

---

1. FEED-FORWARD GAINS (OR INTEGRAL TERMS) ENSURE DISTURBANCE REJECTION.
2. THRESHOLD PREVENTS FLICKERING DUE TO NON-LINEAR DYNAMICS.
3. PERMUTATIONS ARE COMPUTED BEFOREHAND (SEE OFFLINE ALGORITHM IN FIG.15).
4. ROLE ASSIGNMENT IS REPRESENTED BY MATRIX $\hat{\pi}$ FOR CONVENIENCE, YET IMPLEMENTATION RELIES ON A SIMPLE INDIRECTION RATHER THAN MATRIX MULTIPLICATION.

FIG.7A

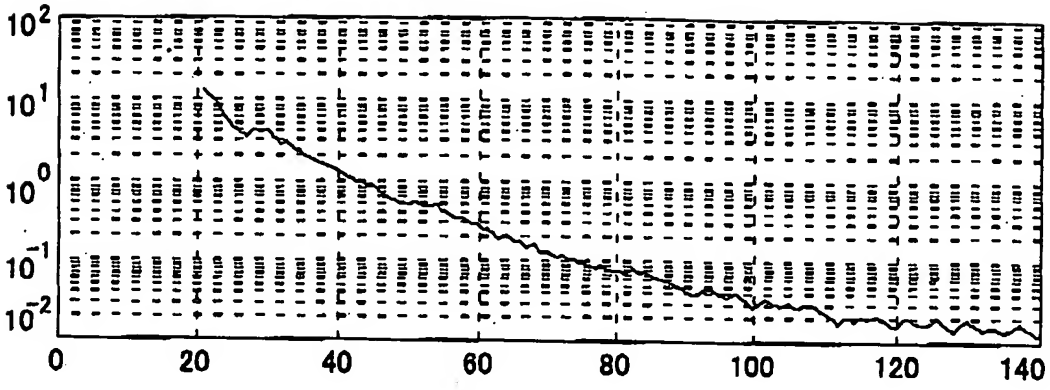

PERFORMANCE INDEX
10^2
10^1
10^0
10^{-1}
10^{-2}
0
20
40
60
80
100
120
140

FIG.7B

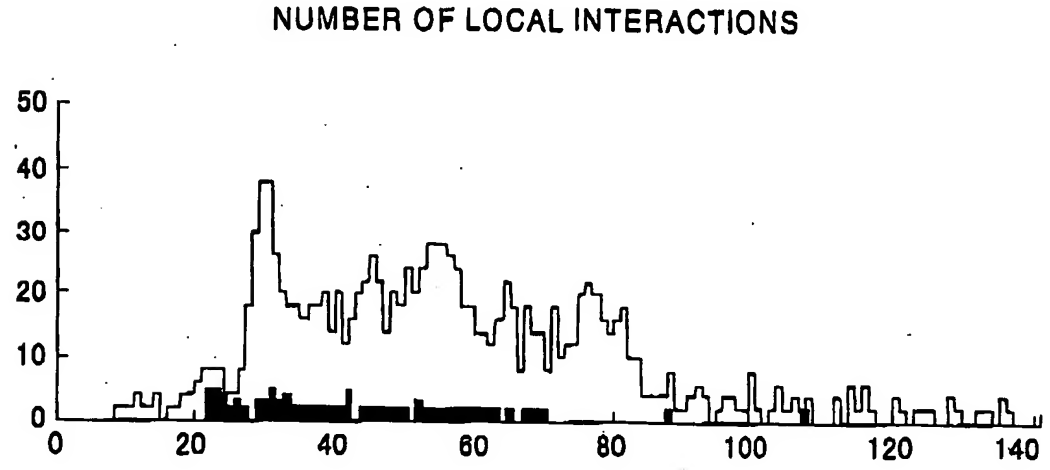

NUMBER OF LOCAL INTERACTIONS
50
40
30
20
10
0
0
20
40
60
80
100
120
140

FIG.8

OPTIMIZED VERSION FOR BLOCK-DIAGONAL SYSTEMS (A2)

```
1     U_n ← -G[X_n - π_n . X_ref]
2     X_n+1 ← f(X_n, U_n)
3     For j=1 to N do
4       δJ(i,j) ← [X_n^j - (π_n . X_ref)^j]^T S^entity [X_n^j - (π_n . X_ref)^j]
5     endfor
6     Sort entities by decreasing δJ(i,I) and select first K ones
7     for k=1 to K do
8       for p=k to K do
9         δJ(k,p) ← [x_n^K - x_ref^p]^T S^entity [x_n^K - x_ref^p]
10      endfor
11    endfor
```

LOOK FOR OPPORTUNISTIC ROLE SWAPS IN K WORST PERFORMERS

```
12    For K! permutations π(k) of K candidates do
13      J_∞(π(k)) ← Σ_p=0^N δJ(p, -x_ref(p,π(k)) or target(p, π(k)))
14      if (J_∞ > | J_∞(π(k)) + Δ_min |) then
15        J_∞* ← J_∞(π(k))
16        k* ← k
17      endif
18    endfor
19    π_n+1 ← π(k*)
```

(1) SORTING CANDIDATES FOR ROLE SWAPPING ON A SIMPLE DISTANCE-TO-TARGET CRITERION CORRESPONDS TO REPLACING S^{entity} BY THE IDENTITY MATRIX.

FIG.9

MEMBER	ALGORITHM	N	n_e	Nn_e	COORDI-NATION	TOTAL/ STEP	REAL-TIME
FISH	A2	6	6	36	8KFLOPS	16KFLOPS	YES
AIRCRAFT	A2	6	11	66	14.6KFLOPS	23KFLOPS	YES
BEE	A1	50	4	20	17MFLOPS	16.8MFLOPS	RESTRICTIVE
BEE	A1	100	4	400	67MFLOPS	68MFLOPS	NO

FIG.10

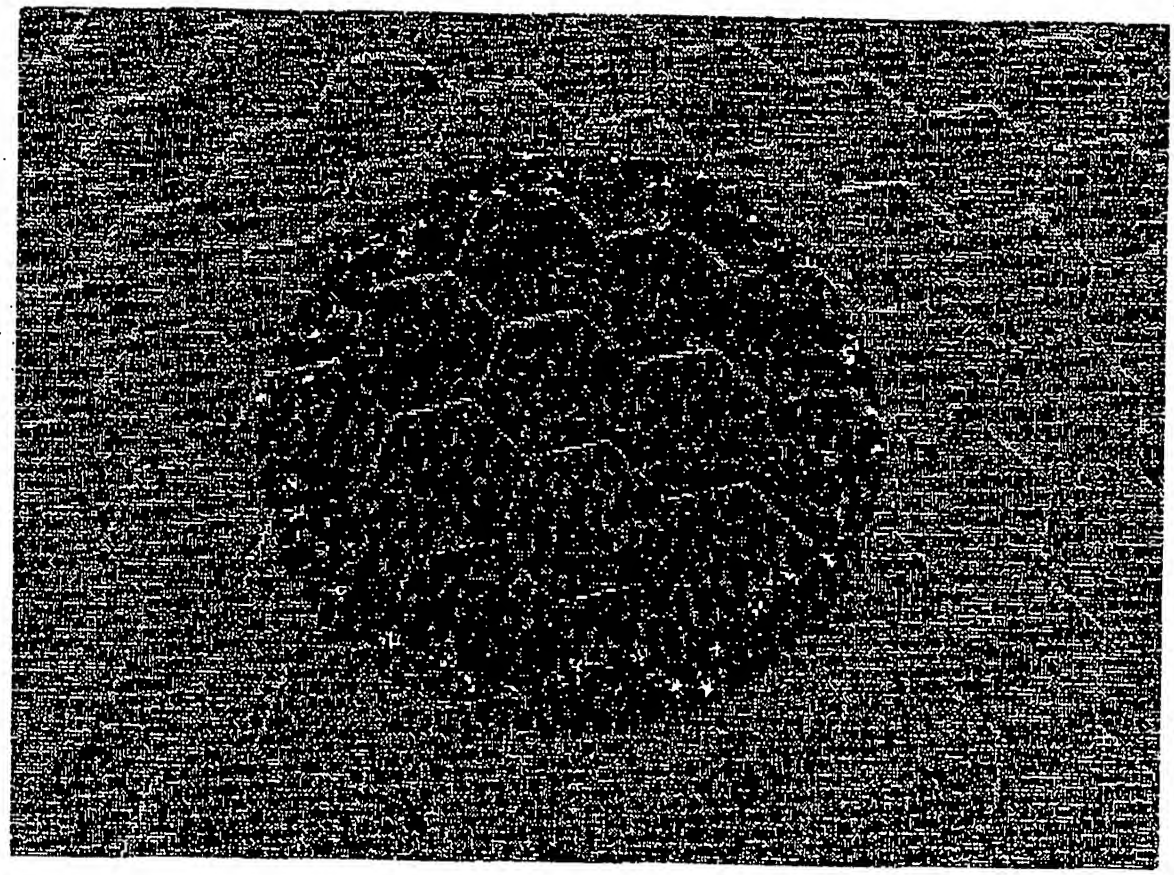


FIG.11

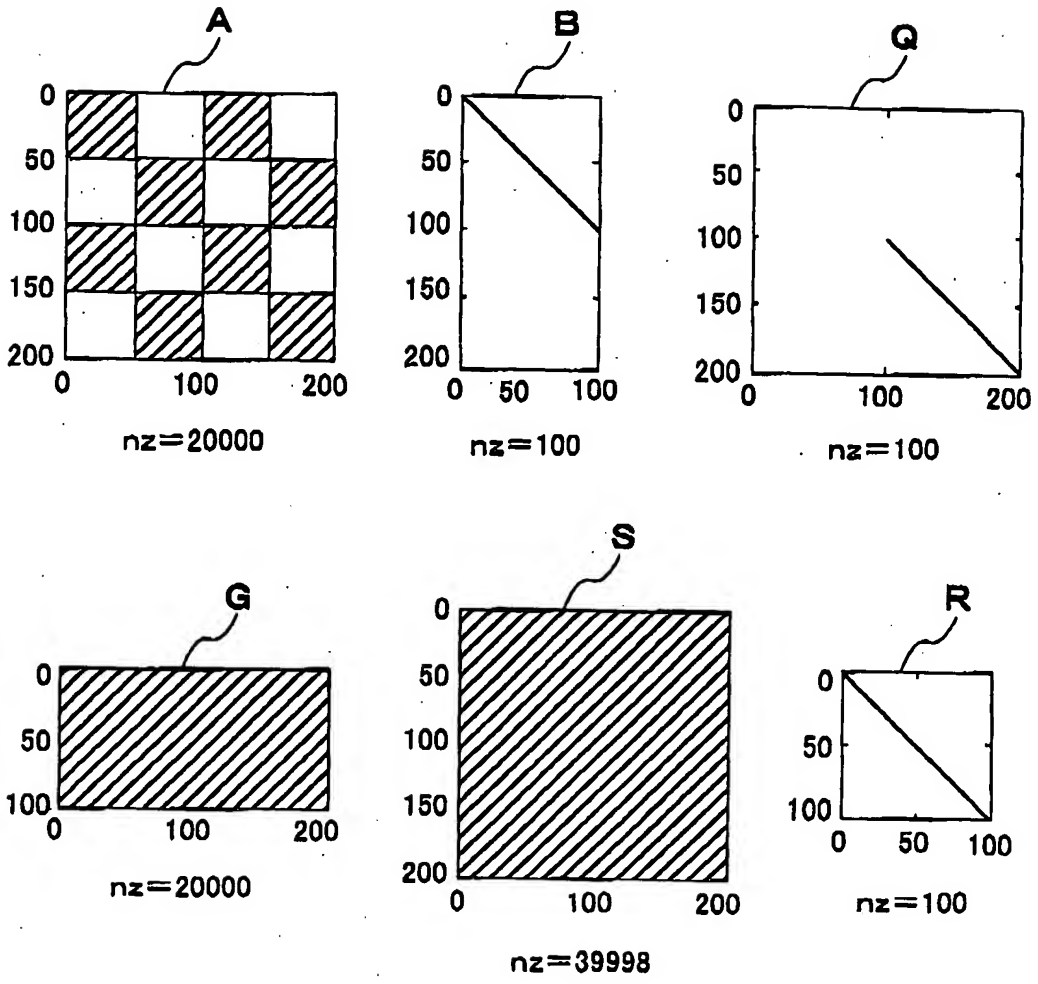

A
0
50
100
150
200
0
100
200
nz=20000
B
0
50
100
150
200
0
50
100
nz=100
Q
0
50
100
150
200
0
100
200
nz=100
G
0
50
100
0
100
200
nz=20000
S
0
50
100
150
200
0
100
200
nz=39998
R
0
50
100
0
50
100
nz=100

FIG.12A

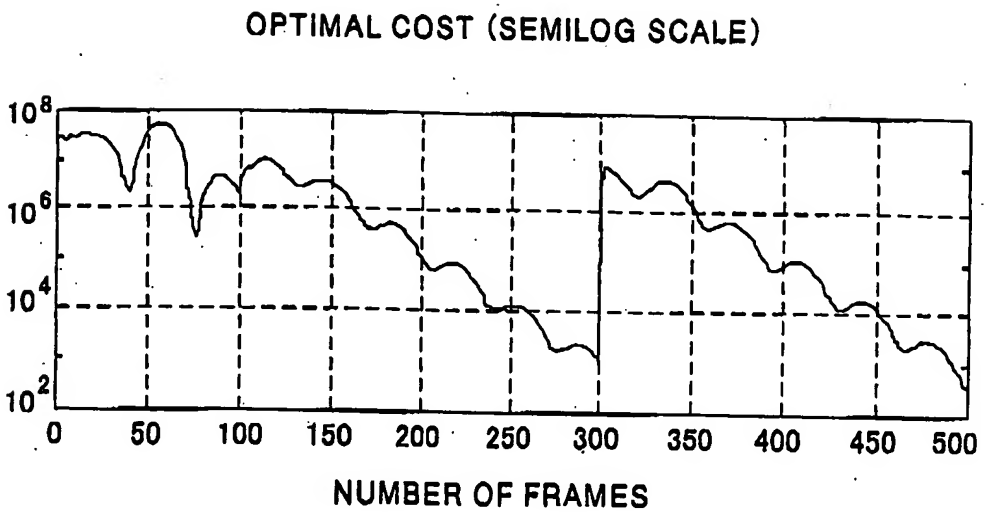


FIG.12B

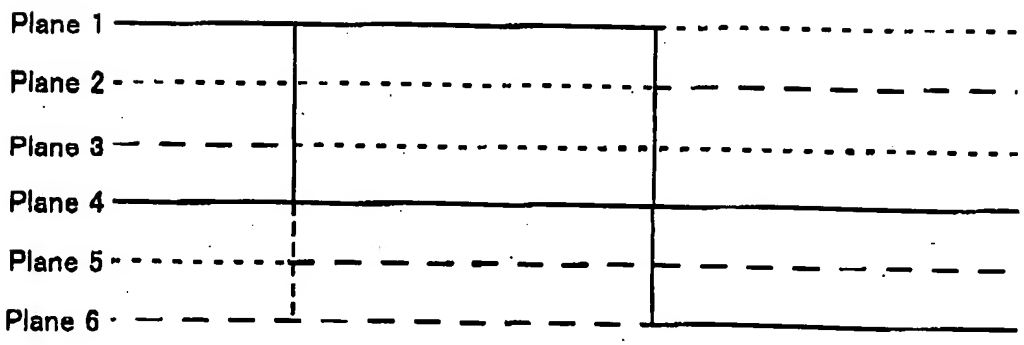


FIG.13A

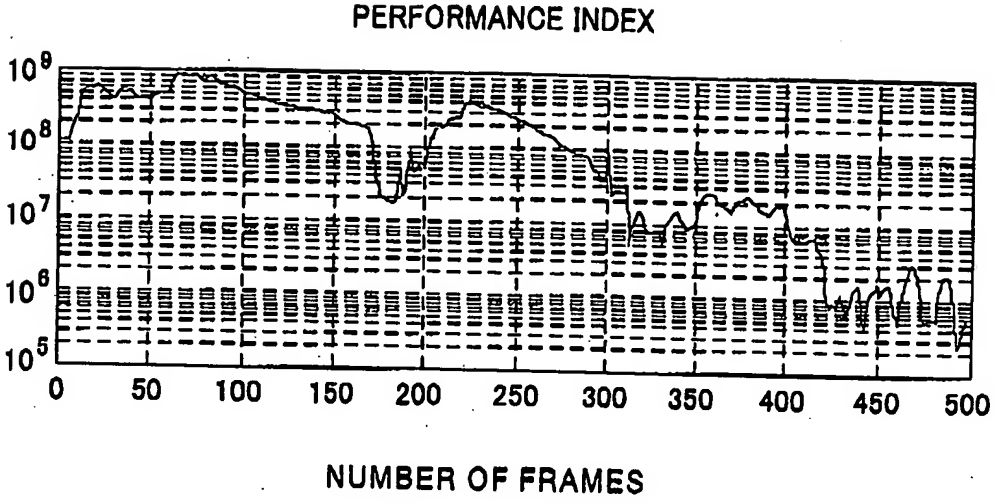


FIG.13B

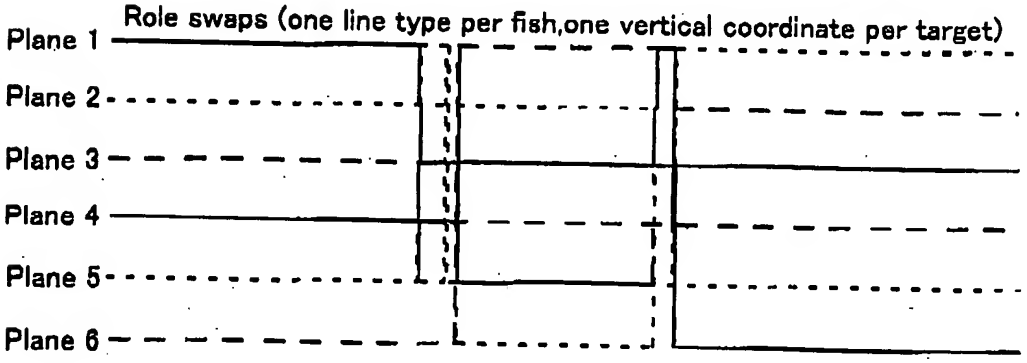


FIG.14

MULTIVARIABLE LQ CONTROL SYSTHESIS

1 GIVEN THE DYNAMIC SYSTEM A, B AND PERFORMANCE INDEX MATRICES Q, R, FORM THE HAMILTONIAN MATRIX DEFINED AS:

$$H=\begin{bmatrix} B+AR^{-1}B^{T}A^{-T}Q & -BR^{-1}B^{T}A^{-T} \\ -A^{-T}Q & A^{-T} \end{bmatrix}$$

2 COMPUTE EIGENVALUES λ AND EIGENVECTORS W OF H, COLLECTING ALL COUPLES SUCH THAT THE EIGENVALUE'S MODULUS IS SMALLER THAN ONE (STABLE POLE OF THE CLOSED-LOOP SYSTEM)

3 COMPUTE $S_{\infty}=\lambda W^{-1}$

4 COMPUTE THE OPTIMAL MATRIX K GAIN AS

$$G=[R+B^{T}S_{\infty}B]^{-1}B^{T}S_{\infty}A$$

FIG.15

```
Function PermutationArray=Permute(vector)
1       n=length(vector)
2       if(n==1) then
3         PermutationArray=vector
4       else
5         PermutationArray=NULL array
6         Height=0
7         for k=1 to n do
8           sub-vector(k)=ShrinkVector(vector, k)
9           SubArray=Permute(sub-vector)
10          for i=1 to (n-1)! do
11            for j=1 to n-1 do
12              PermutationArray [i+Height][j]=SubArray[i][j]
13            endfor
14            PermutationArray [i+Height][n]=vector [k]
15          endfor
16          Height=Height+(n-1)!
17        endfor
18      endif
19      return PermutationArray
```

FIG.16

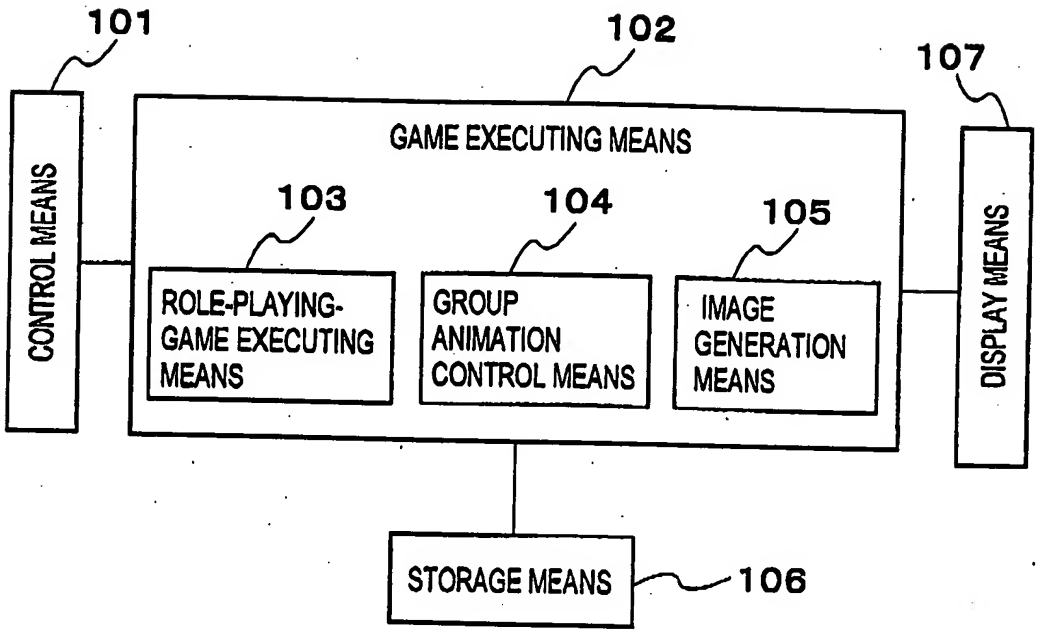

101
102
107
CONTROL MEANS
GAME EXECUTING MEANS
103
104
105
ROLE-PLAYING-GAME EXECUTING MEANS
GROUP ANIMATION CONTROL MEANS
IMAGE GENERATION MEANS
DISPLAY MEANS
STORAGE MEANS
106

FIG.17

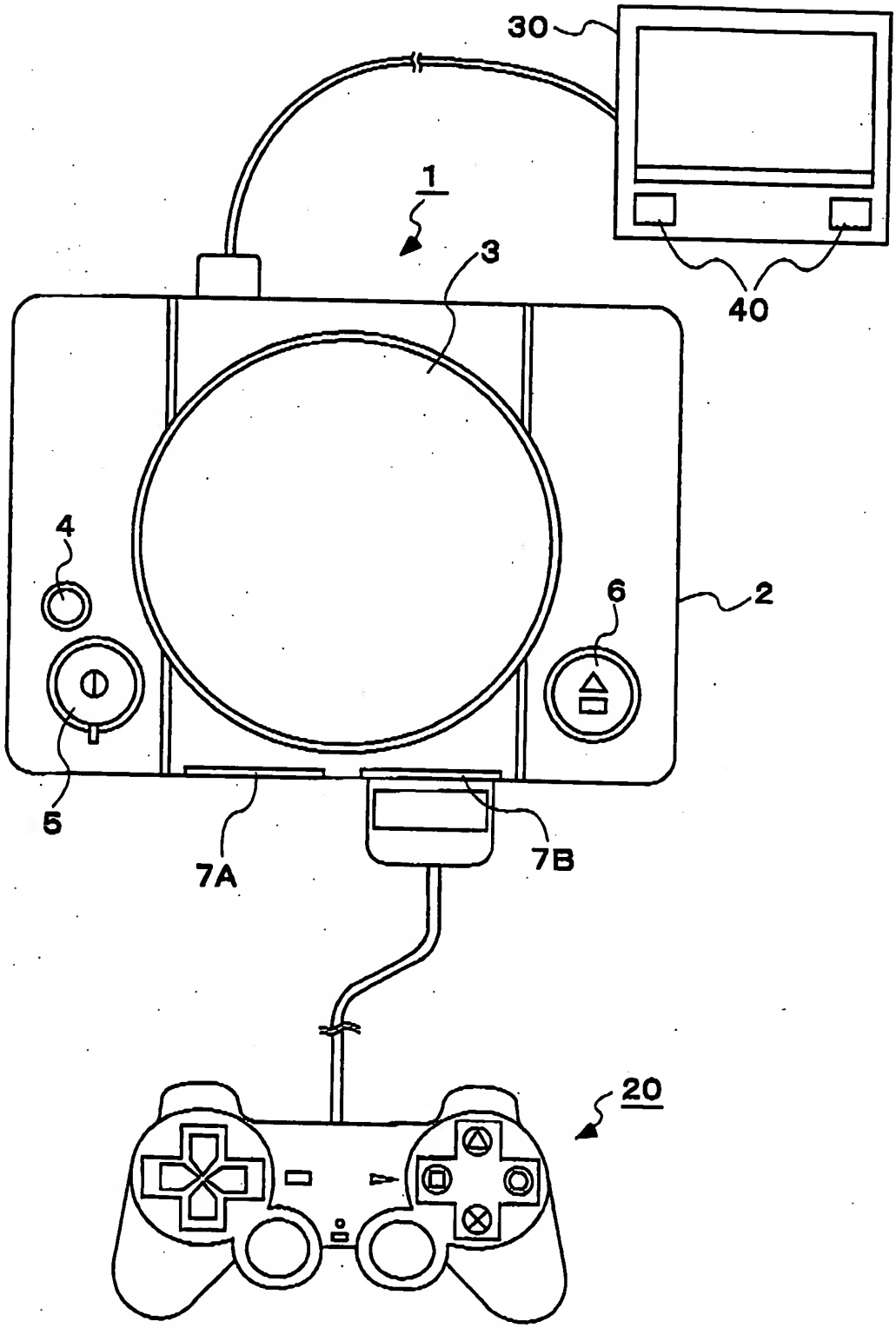
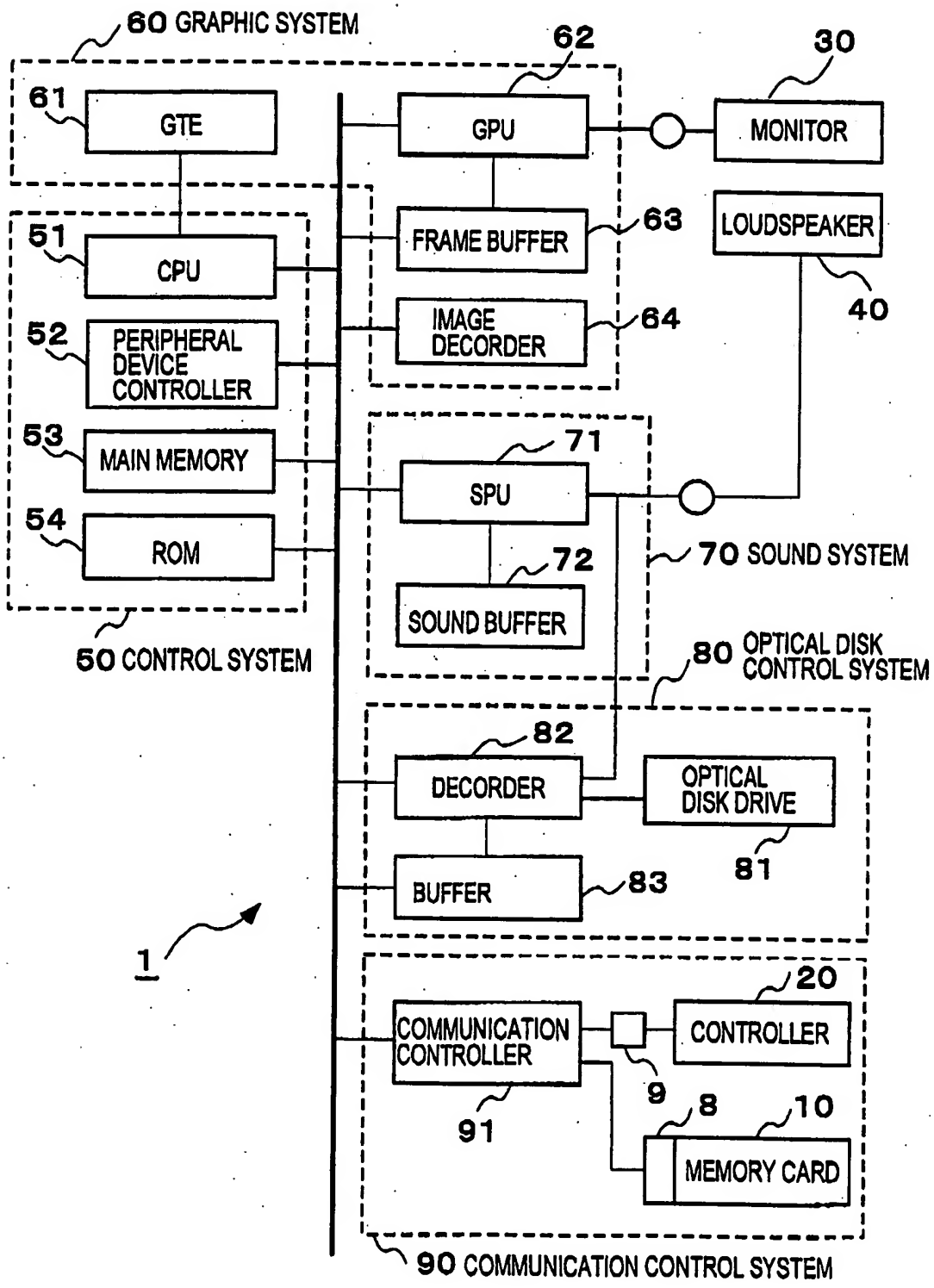


FIG.18


60 GRAPHIC SYSTEM
62
30
61
GTE
GPU
MONITOR
FRAME BUFFER
63
LOUDSPEAKER
51
CPU
IMAGE DECORDER
64
40
52
PERIPHERAL DEVICE CONTROLLER
53
MAIN MEMORY
71
SPU
54
ROM
72
70 SOUND SYSTEM
SOUND BUFFER
50 CONTROL SYSTEM
80 OPTICAL DISK CONTROL SYSTEM
82
DECORDER
OPTICAL DISK DRIVE
81
BUFFER
83
1
20
COMMUNICATION CONTROLLER
CONTROLLER
9
91
8
10
MEMORY CARD
90 COMMUNICATION CONTROL SYSTEM